Pergamon

Computers Math. Applic. Vol. 32, No. 1, pp. 93–96, 1996
Copyright©1996 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0898-1221/96 $15.00 + 0.00

S0898-1221(96)00092-2

Software for the Mathews Stanford Radio Drum

G. W. LOGEMANN
21 Terry's Plain Road
Simsbury, CT 06070-1814, U.S.A.
70641.1421@compuserve.com

Abstract—The Mathews Radio Drum is explored as a multidimensional controller for music. The techniques can be applied to any three-dimensional spatial system. Some problems are examined regarding notation and interpretation of gestures with respect to composition, and the organization of hardware and software for real-time interactive performance.

Keywords—Multidimensional controllers, Radio Drum, Polhemus Isotrak, Spatial gestures, Interactive composition, Conducting systems, Active listening, Carnegie Mellon Toolkit, Gestural notation.

1. RADIO DRUM HARDWARE

The Mathews Radio Drum is one of an expanding family of nonconventional devices for controlling MIDI performances. The unit used in the work to be described is a preliminary model built by M. Mathews of Stanford University. The drum has a surface 18 × 21 containing antennas generating changing voltages received from radio transmitters within two drum sticks called "batons." In my system, these voltages are analyzed with a Data Translation DT2814 multiplexing A/D converter installed in a host IBM AT lookalike, a Chaplet LA-30A laptop with an Intel 80286. The Chaplet also has a Voyetra 4001 MIDI interface, a Roland MPU-401 lookalike.

2. DRUM SOFTWARE

Primary software for the Radio Drum is the Conductor Program written by Mathews. Its principal objective is the performance of Western Euroamerican classical and current (WECC) art music, scores including any such as musical theater and pop that admit standard notation, a process Mathews calls "active listening."

Through active listening, an even less-than-skilled amateur can construct a performance of WECC music by directly controlling some parameters, nominally the nonpitch components, e.g., tempo, loudness, and expression. Personally, I find active listening a very satisfying educational experience. I believe the technique can be useful in teaching all students, not just those interested in pursuing music. I think an important use of active listening will involve performing in conjunction with other live musicians, e.g., instrumentalists, singers, and dancers. Finally, I find active listening generally useful in composing and specifically in developing the live component of interactive music.

I thank R. Boulanger for reading a preliminary version of this paper at the SEAMUS 93 conference and answering questions as needed. Any problems that arise are mine; please feel free to contact me directly. Also, I wish to thank M. Mathews for supplying my preliminary Drum and software. And finally, I thank R. Dannenberg for years of conversations, including helpful comments regarding this paper.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

The score, with notes, drum strokes, and control information, is represented in a more-or-less mnemonic alphanumeric language. Drum outputs are MIDI messages that can be sent to a synthesizer. Alternatively, I have written interactive compositions that use a second LA-30A to interpret the MIDI messages and compute responses.

Internally, the principal task of Conductor Program is to interpret the Drum voltages as "whacks" and other "control values."

A "whack" occurs when one or the other baton strikes the surface. A "control value" is an x or y position relative to the left-right or forward-backward dimension of the surface. A "whack" generally triggers one or more MIDI messages which are usually note-on/note-off pairs but can be, e.g., program changes; control values from either the whacking or nonwhacking baton can determine the velocity and duration of the notes. Other messages generated from a whack can be program change or MIDI controller messages; alternatively, the x-y control values can be sent continuously to one or another MIDI controller. Tempo changes can be triggered by whacks or tempo can be taken continuously from x-y values, and so on.

An interesting problem is the calculation of implied tempo based on when whacks are to be expected versus when the system detects them from the Drum.

Another interesting aspect of the technical calculations is the "calibrating" of the Drum, parameters used to convert the A/D values into coordinates x and y that are fairly linear and independent along the surface.

2.1. Enhancements to Conductor Program

Having performed a few WECC scores, one can be impressed with the potential of the Drum in active listening and in performing generalizations of current WECC forms (cf. the works of R. Boulanger). Via such generalizations, the Drum can be compared to various devices presented by several researchers at recent SEAMUS and ICMC conferences, such as conducting batons and gloves at various levels and spatial sensors such as the Polhemus Isotrak. Herein I describe some independent experiments in this direction.

My first experiments were to employ the z-direction, perpendicular to the Drum surface, and more general gestures (than whacks) as triggers and more general movements (than along the surface) as sources for control values.

Dependency on z was introduced in several ways. It was useful to change notation: the vertical dimension was already calculated in Conductor Program, but in a value I changed to be called w (large near the surface and decreasing to 0 moving upward away from the surface). It is amusing to note that, for suitable a, b, and c, the relationship

$$z = \frac{a}{w - b} + c$$

yields a value of z increasing linearly over a useful range above the surface. Not only does the inverse relationship show a similar dependence of w on z, but given three points (w_1, z_1), (w_2, z_2), (w_3, z_3), one surprisingly can solve explicitly for a, b, and c in closed form, yielding easy calibration.

It is relatively easy then to use z itself as a control value. An obvious addition is to use v, the velocity of the baton at the surface just prior to the moment of whack, as a control value.

Not so obvious, but extremely sensitive and effective, is to use m, the maximum of z computed during the entire whack trajectory, as a control value. In some sense, m seems superior to v.

Other than control values, z was analyzed for "upwhacks," turning around at an indefinite level above the surface. Upwhacks seem to be useful when (down)whacks must occur quickly, e.g., to establish a tempo, or when an upbeat is natural. These concepts can be pursued further, e.g., the maximum height or average speed during an upwhack; a downwhack to a point above the surface whose distance yields a control value, etc.

The problem here, however, becomes one of notating the different flavors (to coin a term) of whack. Also, one needs to establish a series of precedence rules. For example, what is to be done if a downwhack is to be expected next in the score but the system detects an upwhack? As a generalization of the precedence rules, there are a number of options that need to be turned on and off at various times; solving this problem is a question of algorithm as well as notation.

In addition to dependency on z, other calculations were explored. One seemingly useful change involves generating MIDI controller messages (to change the setting in the synths) only when control values change by a significant amount, thus reducing MIDI congestion.

2.2. Other Software than the Conductor Program

In order to facilitate experiments in active listening, it seemed it would be useful to admit a more generalized notation for scores and to allow MIDI files as input data. Also, for interactive composition. it seems clear that a good deal more algorithmic decision making is necessary, e.g., to support the new flavors of whack, precedence, and calculation of triggers and control values. To this end it seemed useful to explore other notation and programming systems.

An obvious choice is the Max programming language. This may indeed turn out to be the best choice (cf. the work by A. Schloss). However, although my experiments with the Max system show it to be extremely powerful and facile within its vocabulary of primitives, e.g., for descriptions of control and flow, its primary drawback to me is my prior training in procedural languages, such as C, for developing new algorithms. Therefore I have spent considerable time adapting R. Dannenberg's Carnegie Mellon Toolkit (CMT) to the needs of the Drum.

An interesting aspect of CMT is its two major components of a notation language, Adagio, and a real-time C programming environment, MoxC. It is extremely interesting that these two are now integrated, so that Adagio sequences can call C routines (quasi "objects") and MoxC programs can load, trigger, and modify high-level notation sequences (written in Adagio).

A routine was written, drawing on MoxC routines, to translate Adagio sequences into Conductor Program performance scores. For free, a byproduct of this translation program is the ability to create Conductor Program performance scores from MIDI files.

The translation requires some method to indicate to the system various "declaration" options, e.g., which control values are to be linked to noteon velocity or specific MIDI controller messages: Adagio has a macro facility that permits creating MIDI system exclusive (sysex) messages that are thereby rendered transparent to synthesizers. This last feature is useful in that whack triggers are coded as notes for a particular MIDI channel; if one uses a metronome click for such notes, one can audit the whacks as the sequence is played, without interference from the sysex information.

2.3. Future

This effort is still underway. A useful target project is to integrate the active listening (score realization) process with the MoxC interpretation routines, so that the functions of the two LA-30A Chaplets can be carried out in concurrent processes on the same machine. A second project is to create or simply edit the "whack track" from gestures directly from the Drum. A third project is to explore the full spectrum of three- (or higher-)dimensional gestures, also useful to gloves and Polhemus sources. Collateral projects, not directly involving but impinging on the Drum, are a MIDI file editor and the incorporation of a non-MPU MIDI interface using the IBM PC's serial ports, thus allowing MIDI musicians to perform with IBM laptops and notebooks.

REFERENCES

1. R.B. Dannenberg, *The Carnegie Mellon Toolkit*, Carnegie Mellon University, Pittsburgh, PA, (1986).
2. G.W. Logemann, Experiments with a gestural controller, In *Proceedings of the 1989 ICMC*, pp. 184–185, Computer Music Association, San Francisco, CA, (1988).
3. G.W. Logemann, Cascades: Algorithmic gestural performance, *SEAMUS* **VI** (1), 10–15 (1991).

4. G.W. Logemann, Cascades: Interactive gestural performance, In *Proceedings of the 1991 ICMC*, pp. 401–403, Computer Music Association, San Francisco, CA, (1991).
5. M. Pluckette and D. Zicarelli, *Max*, Opcode Systems, Inc., Menlo Park, CA, (1991).